Reading Data from the Web

Week 2 - Getting an Cleaning Data

Krystella

2/11/2022

```
knitr::opts_chunk$set(echo = TRUE, eval = FALSE)
```

Webscraping

- Programatically extracting data from the HTML code of websites
- It can be a great way to get data; many websites have data you may want to programatically read
- In some cases this is **against the terms of service** for the website - some websites do not explicitly say they do not want their data to be read
- Attempting to read too many pages too quickly **can get your IP address blocked**

Getting Data Off Webpages using readLines()

```
con = url("http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en")
htmlCode = readLines(con)
close(con)
htmlCode
```

Parsing with XML

```
library(XML)
url <- "http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en"
html <- htmlTreeParse(url, useInternalNodes=T)

xpathSApply(html, "//title", xmlValue)
```

```
xpathSApply(html, "//td[@id='col-citedby']", xmlValue)
```

GET from the httr package

```
library(httr); html2 = GET(url)

content2 = content(html2, as="text")
parsedHtml = htmlParse(content2, asText = TRUE)
xpathSApply(parsedHtml, "//title", xmlValue)
```

Accessing Websites with Passwords

```
pg2 = GET("http://httpbin.org/basic-auth/user/passwd",
          authenticate("user", "passwd"))
pg2
```

```
names(pg2)
```

Using Handles

```
google = handle("http://google.com")
pg1 = GET(handle=google, path="/")
pg2 = GET(handle = google, path="search")
```

Further Resources

- R Bloggers : http://www.r-bloggers.com/?s=Web+Scraping
- httr help file : http://cran.r-project.org/web/packages/httr/httr.pdf